基于 YOLO 算法的标签识别检测

吕泽钰
（北京化工大学 机研 2301）

1 前言

随着人工智能的兴起与信息技术的不断发展，自动化、智能化已经是工业界生产革命的必然方向。针对传统 PET 瓶生产线存在的检测问题，提出了一种新颖的，实用性极高的自动化架构，结合现有的实验室条件搭建了测试平台。

在视觉的实现方面，使用 YOLOV5 算法，通过自主收集 PET 瓶、制作标签获得样本集。训练用样本总数为 1070，其中测试集数目 107，训练集数目 963。

2 视觉模型——YOLO 算法

视觉检测部分是自动化生产线中的关键环节，目的是正确且高速地识别出待检 PET 瓶的生产质量。较为常用目标检测方法有模板匹配法、R-CNN 算法、Fast R-CNN 算法、YOLO 算法、SSD 算法等。由于本文研究的是一个工业生产领域的实际运用，对生产效率有其内在要求，因此选择 YOLO 算法作为视觉检测算法，其快速高效的特点十分契合现代化工业生产。

YOLO 全称为 You Only Look Once，是 Joseph Redmon 于 2015 年提出的一种目标检测框架。相对于最为经典的 RCNN 系列算法，YOLO 算法以其独特的方式处理目标检测：通过将整个图像放在一个实例中，并预测这些框的边界框坐标和及所属类别概率。它的最大特点正如其名：把目标检测当做一个单一的回归任

务，因此拥有极快的响应速度，受到工业界欢迎。

最初的 YOLOV1 是将 416×416×416 的图片，分成 7×7×7 的网格，最终预测的是一个长度为 S×(B×5+C)的向量，其中 S=7,B=2,C 是预测的类别数。

YOLOV2 对一代进行了小改进，作者参考了使用多尺度的特征图做检测的 SSD，提出了 pass through 层从而实现多尺度检测。和 V1 相比，在使用 Fine-Grained Features 时，YOLOV2 的性能提升了 1%。YOLOV3 是在当时 fpn 兴起的背景下产生的，因为小目标经过多层卷积之后可能会丢失特征或变得微小，所以考虑将浅层和深层特征组合在一起，从而获得表层特征和语义特征，这种方法在 YOLOV3 中表现出良好的性能。

YOLOV4 的大体框架基于 V3，且在先版本的基础上采取了一些改进，例如在 CSPDaknet53 上添加 SPP 块、使用 PANet 取代 YOLOV3 中的 FPN 进行参数聚合等等。除了改进骨干网络，YOLOV4 在数据增强方面进行了创新，例如引入了 Mosaic 和 Mixup 等方法。Mosaic 是将四张图片拼接成一张图用于目标检测，Mixup 则是将两个图像相加，这两种方法具有背景信息丰富的优点，可以有助于对象检测。此外，在 neck 方面，它使用了 PANet 的方法，不仅采用了两次上采样 concat 的方式，还在此基础上使用了两次下采样，以更有效地融合特征。

YOLO 算法的核心思想是利用整张图片作为网络输入，并直接在输出层回归 bounding box 的位置和所属类别。

首先，YOLO 的 CNN 网络将输入的图片分割成 S×S 网格，每个单元格预测 B 个边界框及其置信度 $\mathrm{Pr(object)}$。其准确度记为 $\mathrm{IOU}_{\mathrm{pred}}^{\mathrm{truth}}$，也就是实际框与预测框的交并比。因此将置信度定义为：

$$Truth = Pr(object) \times IOU_{pred}^{truth}$$

边界框的位置与大小用(x, y, w, h)来记录，其中(x, y)代表边界框的中心坐标（相对于每个单元格左上角坐标点），w 和 h 则分别代表边界框宽和长（相对值），4 个元素的大小都应该保持在 0-1 的范围内。最后可以用(x, y, w, h, c)来表征边界框的预测值，c 代表置信度。

特别的，在分类问题中边界框的每个单元格其要给出预测的 C 个类别概率值，记为 $Pr(class_i|object)$，则各个边界框类别置信度为：

$$Pr(class_i|object) \times Pr(object) \times IOU_{pred}^{truth} = \times Pr(class_i) \times IOU_{pred}^{truth}$$

YOLO 使用了卷积网络模型进行特征值提取，其卷积层数为 24、全连接层数为 2。它主要使用 1×1 卷积紧跟 3×3 卷积作为全连接层。对于卷积层和全连接层，采用激活函数 $\max(x, 0.1x)$，最后一层采则用线性激活函数。

对于 loss 函数，yolo 的设定如下：

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2]$$
$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\hat{h_i}})^2]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C_i})^2$$
$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C_i})^2$$
$$+ \sum_{i=0}^{S^2} \sum_{c \in classes} (p_i(c) - \hat{p_i}(c))^2$$

公式的第一项代表边界框中心坐标的误差项，1_{ij}^{obj} 指的是第 i 个单元格存在目标，且该单元格中的第 j 个边界框负责预测该目标；公式的第二项是边界框的高与宽的误差项；第三项是包含目标的边界框的置信度误差项；第四项是不包含目标的边界框的置信度误差项；最后一项则是包含目标的单元格的分类误差

项， 1_i^{obj}指的是第 个单元格存在目标；对于置信度值 Ci，一般取为 1。

在实现预测时候，可能会出现多个预测值，这时候就要使用非极大值抑制算法选出最后结果。论文最后选择使用 yolov5 版本完成视觉部分，其结构框图 1 所示。

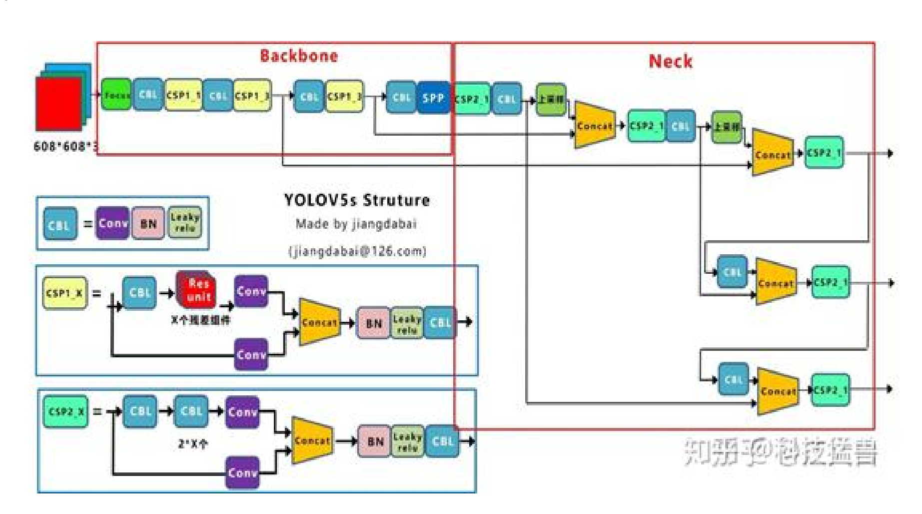


图 1 YOLOV5 结构图

3 检测样本预处理

将采集所得图片使用 labelimg 进行标注，划分了 5 种标注样本 :Label_have，Label_loss，Label_broken，Cap_ture，Cap_false，分别对应贴标存在，贴标缺失，贴标破损，瓶盖正常，缺失瓶盖。

在完成了样本收集和数据标注之后，得到了 1070 个 xml 格式的样式文件，进行预处理。

（1）将文件的重命名，标签和图片需要按照一定的格式顺序进行命名。

```
@echo off&setlocal EnableDelayedExpansion
set a=1
for /f "delims=" %%i in ('dir /b *.jpg') do (
if not "%%~ni"=="%~n0" (
if !a! LSS 10 (ren "%%i" "yb00!a!.jpg") else if !a! LSS 100 (ren "%%i" "yb0!a!.jpg" ) else ren "%%i" "yb!a!.jpg"
set/a a+=1
)
)
```

图 2 命名代码

（2）标签格式的转换

```
<annotation>
	<folder>JPEGImages</folder>
	<filename>sp006.jpg</filename>
	<path>F:\同步空间\毕业设计\代码\VOC2007\JPEGImages\sp006.jpg</pat
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>500</width>
		<height>488</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>Cap_ture</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>131</xmin>
```


sp008.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V)


```
0 0.504 0.244 0.168 0.18
3 0.503 0.533 0.326 0.758
```


图 3 转换前的 xml 格式（左）与转换后的 txt 格式（右）

（3）Yolov5 需将数据划分为训练集和测试集，本文采用均匀分布方式抽样，划分训练集与测试集的比列为 9：1。最终样本的使用情况为：测试集 107 个，训练集 963 个。

4 模型训练与效果

综合考虑性能和训练时间之后，选择 YOLOV5s.pt 作为预训练权重以保证模型的检测精度和检测速度。

编写 yb.yaml 文件，令其中 nc=2，names: ['Cap_ture', 'Cap_false', 'Label_have', 'Label_loss', 'Label_broken']。选定训练轮次 300。

表 1 训练参数

参数	value	名称	参数
models	yolo5s.yaml	lr0	0.01
resume	Ture	lrf	0.2
workers	6	momentum	0.937
epochs	300	warmup_epochs	3.0
batch-size	12	warmup_bias_lr	0.1
nc	5	hsv_h	0.015
multi-scale	false	flipud	0.8
save-period	1	fliplr	0.5
optimizer	SGD	anchor_t	4.0

计算训练结果的各项数学指标，并生成图像一一罗列。

图 4 是训练结果的 F1 分数与置信度的关系图。F1 分数是分类的一个衡量标准，是精确率和召回率的调和平均数，介于 0-1 之间。一般来说，置信度阈值较低的时候，很多置信度低的样本被认为是真，召回率高，精确率低；置信度阈值较高的时候，置信度高的样本才能被认为是真，因此检测越准确，精准率所在前后两头的 F1 分数比较少。可以看到图中 F1 曲线很“宽敞”且顶部接近 1，说明在训练数据集上表现得很好（既能很好地查全，也能很好地查准）的置信度阈值区间很大。

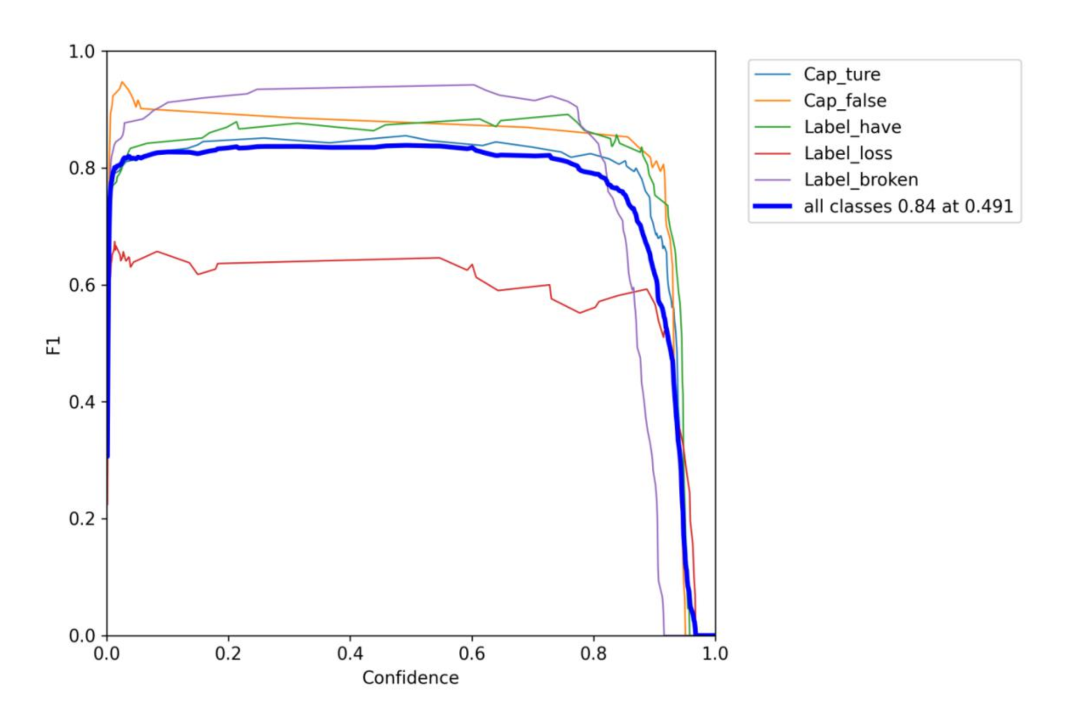


图 4 F1-Confidence 曲线

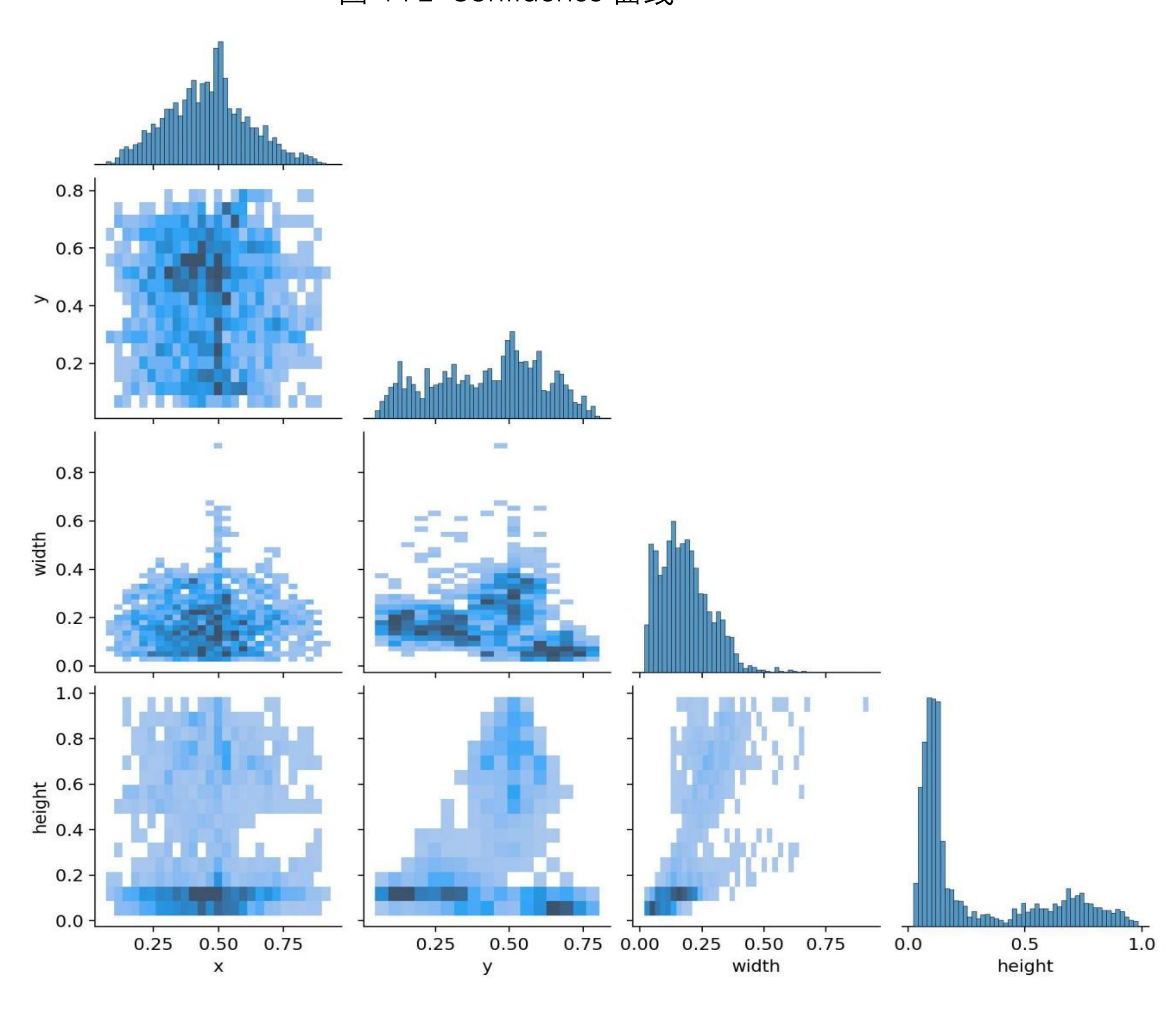


图 5 中心点横纵坐标以及框的尺寸关系

图 5 是标签框中心点坐标与框尺寸关系图。最上面的图表明中心点横坐标 x 的分布情况，可以看到大部分集中在整幅图的中心位置；第一行图表明中心点纵坐标 y 的分布情况，显然大部分点集中在整幅图的中心位置；第二行图表明框的宽的分布情况，本此训练结果框的宽度普遍小于整幅图宽的一半；最后一行图表明框的高的分布情况，可以看到框的高度普遍小于整幅图的高的一半。

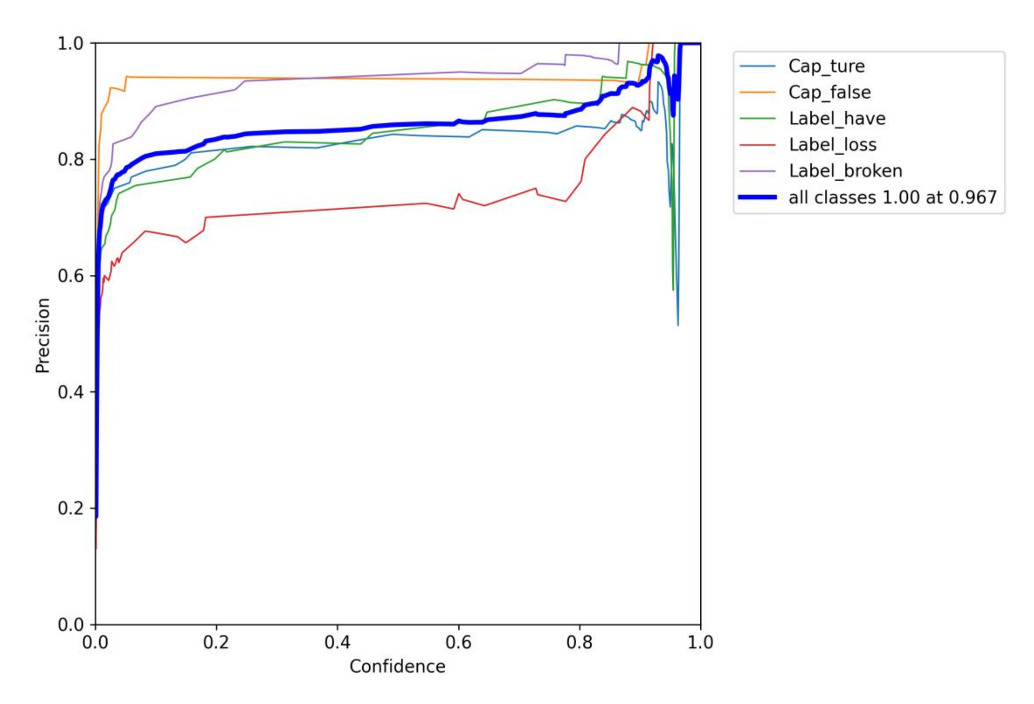


图 6 单一识别精确度

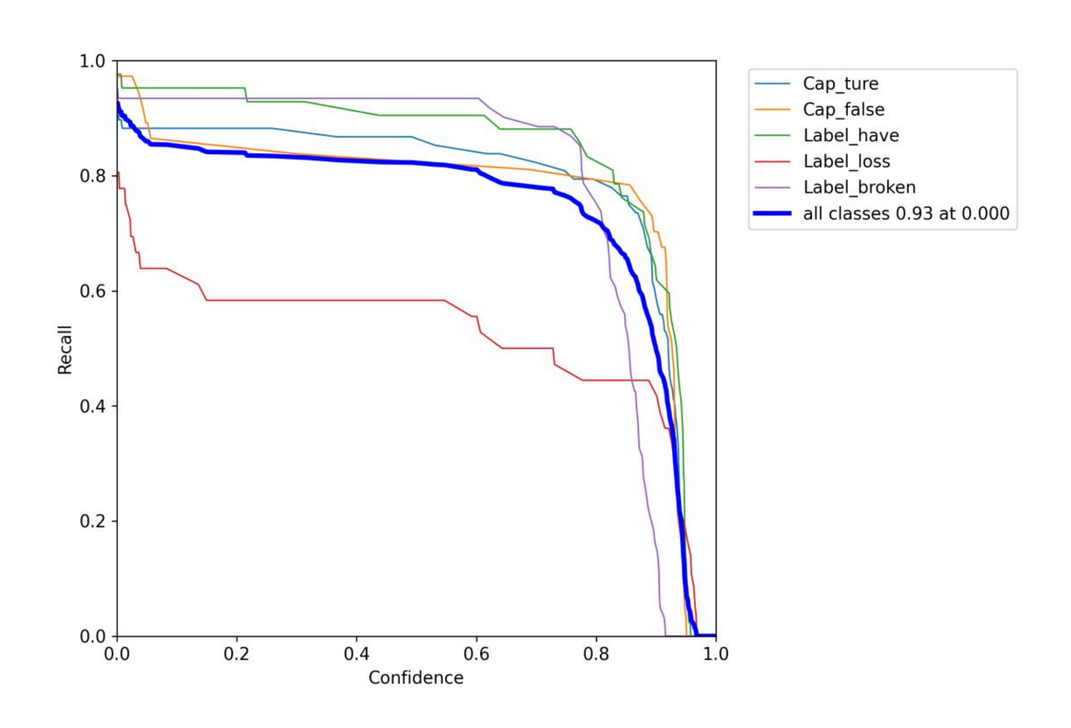


图 7 单一类召回率

图 6 和图 7 分别显示的是各个标签的单一识别精确度 Precision 和单一类召回率 Recall。Precision 衡量的是一个分类器分出来的正类的确是正类的概率，其计算公式为：

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

公式中分子代表找对的正类，分母代表判断为正的所有样本。

Recall 代表召回率，其计算公式为： (3-5)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

由上图可以看出 Precision 保持在 0.9 以上，Recall 保持在 0 附近。

最后的轮次训练结果图如 8 所示。

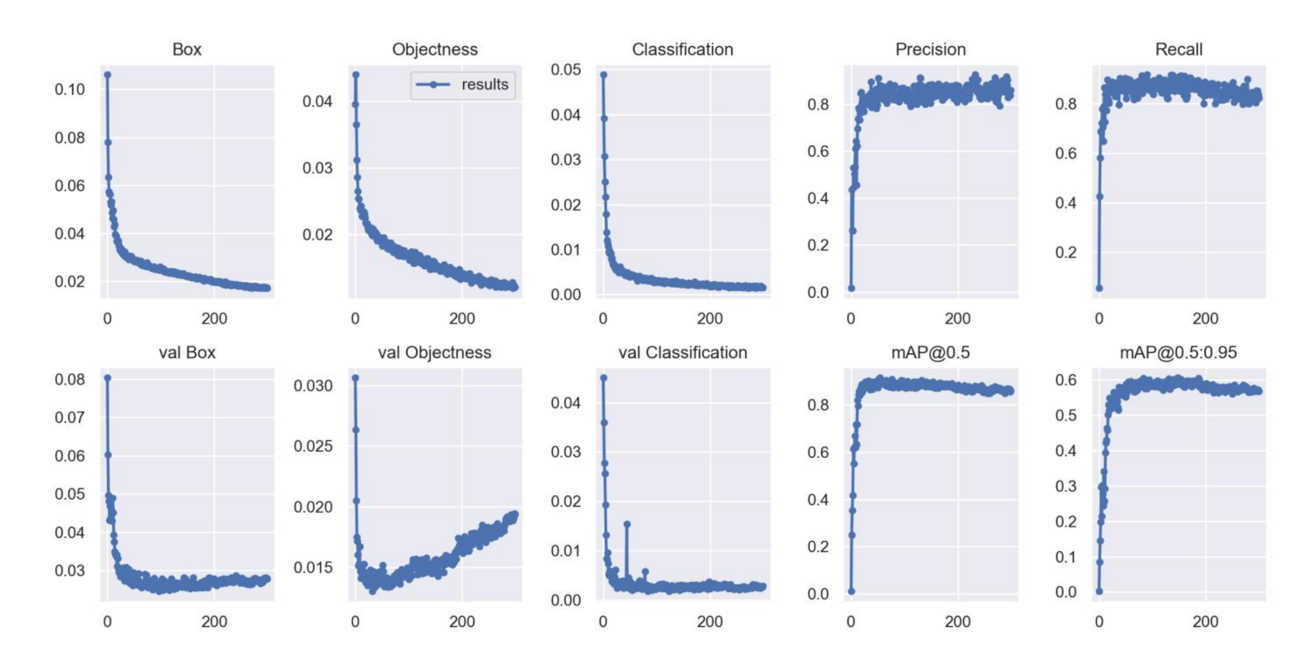


图 8 各指标与轮次关系图

图中 Box 为损失函数均值，可以看出其在 200 轮左右已经达到了理想状态，280 轮时曲线趋势放缓，已经接近 0。

最后两张图是用精确度和召回率两轴作图，m 表示平均，@是标志位，跟随的数字位阈值，作用是判定 iou 为正负样本。例如，@0.5:0.95 表示阈值取 0.5:0.05:0.95 后取均值。

在进行训练指标分析之后，还需要进行实际样本检验。针对 5 种指标，本文分别使用实物样本和网络图片进行实际检验，检验结果见表 2。

表 2 实物检验结果

标签名称	标签数目	监测出的标签数	成功率
Cap_false	24	34	100%
Cap_true	77	77	100%
Label_lose	12	12	100%
Label_broken	33	31	93.7%
Label_have	89	89	100%

可以看出，除了 Label_broken，也就是破损缺陷有少数没有检测出来以外，其余特征的检测成功率均为 100%，没有漏检。





图 9 网络图片效果检验（左）和实物图片效果检验（右）

5 结论

在视觉的实现方面，使用 YOLOV5 算法，通过自主收集 PET 瓶、制作标签获得样本集。训练用样本总数为 1070，其中测试集数目 107，训练集数目 963；在模型训练后的下一步是指标检测与实物样本检测，其中指标检测的精准度和召回

率都达到了 0.9，loss 均值与分类 loss 均值均接近于 0；实物样本检测的最低精确度为 93.7%，部分检验准确率达到了 100%，两种检测都取得了很好的效果。